UNITED STATES PATENT APPLICATION FOR:

NETWORK SURVEILLANCE

INVENTORS:

PHILLIP ANDREW PORRAS
ALFONSO VALDES

ATTORNEY DOCKET NUMBER: SRI/3928-9

CERTIFICATION OF MAILING UNDER 37 C.F.R. 1.10

I hereby certify that this New Application and the documents referred to as enclosed therein are being deposited with the United States Postal Service on March 22, 2004, in an envelope marked as "Express Mail United States Postal Service", Mailing Label No. EV 367981095 US, addressed to: Commissioner for Patents, Mail Stop PATENT APPLICATION, P.O. Box 1450, Alexandria, VA 22313-1450

Linda De Nardi
Signature

Linda DeNardi
Name

March 22, 2004
Date of signature

MOSER, PATTERSON & SHERIDAN LLP
595 Shrewsbury Ave.
Shrewsbury, New Jersey 07702
(732) 530-9404

NETWORK SURVEILLANCE

[0001]   This application is a continuation of U.S. application serial number
10/254,457, filed September 25, 2002, which is a continuation of U.S.
application serial no. 09/658,137, filed September 8, 2000 (now U.S. Patent No.
6,484,203), which is a continuation of U.S. application serial no. 09/188,739,
filed November 9, 1998 (now U.S. Patent No. 6,321,338), where all applications
are herein incorporated by reference.

Reference to Government Funding

Reference to Appendix

[0003]   An appendix consisting of 935 pages is included as part of the
specification. The appendix includes material subject to copyright protection.
The copyright owner does not object to the facsimile reproduction of the
appendix, as it appears in the Patent and Trademark Office patent file or
records, but otherwise reserves all copyright rights.

Background

[0004]   The invention relates to computer networks.

[0005]   Computer networks offer users ease and efficiency in exchanging
information. Networks tend to include conglomerates of integrated commercial
and custom-made components, interoperating and sharing information at
increasing levels of demand and capacity. Such varying networks manage a
growing list of needs including transportation, commerce, energy management,
communications, and defense.

[0006]   Unfortunately, the very interoperability and sophisticated
integration of technology that make networks such valuable assets also make
them vulnerable to attack, and make dependence on networks a potential

-2-

liability. Numerous examples of planned network attacks, such as the Internet worm, have shown how interconnectivity can be used to spread harmful program code. Accidental outages such as the 1980 ARPAnet collapse and the 1990 AT&T collapse illustrate how seemingly localized triggering events can have globally disastrous effects on widely distributed systems. In addition, organized groups have performed malicious and coordinated attacks against various online targets.

Summary

[0007]   In general, in one aspect, a method of network surveillance includes receiving network packets (e.g., TCP/IP packets) handled by a network entity and building at least one long-term and at least one short-term statistical profile from at least one measure of the network packets that monitors data transfers, errors, or network connections. A comparison of at least one long-term and at least one short-term statistical profile is used to determine whether the difference between the short-term statistical profile and the long-term statistical profile indicates suspicious network activity.

[0008]   Embodiments may include one or more of the following features. The measure may monitor data transfers by monitoring network packet data transfer commands, data transfer errors, and/or monitoring network packet data transfer volume. The measure may monitor network connections by monitoring network connection requests, network connection denials, and/or a correlation of network connections requests and network connection denials. The measure may monitor errors by monitoring error codes included in a network packet such as privilege error codes and/or error codes indicating a reason a packet was rejected.

[0009]   The method may also include responding based on the determining whether the difference between a short-term statistical profile and a long-term statistical profile indicates suspicious network activity. A response may include altering analysis of network packets and/or severing a communication channel. A response may include transmitting an event record

to a network monitor, such as hierarchically higher network monitor and/or a network monitor that receives event records from multiple network monitors.

[0010]   The network entity may be a gateway, a router, or a proxy server. The network entity may instead be a virtual private network entity (e.g., node).

[0011]   In general, in another aspect, a method of network surveillance includes monitoring network packets handled by a network entity and building a long-term and multiple short-term statistical profiles of the network packets. A comparison of one of the multiple short-term statistical profiles with the long-term statistical profile is used to determine whether the difference between the short-term statistical profiles and the long-term statistical profile indicates suspicious network activity.

[0012]   Embodiments may include one or more of the following. The multiple short-term statistical profiles may monitor different anonymous FTP sessions. Building multiple short-term statistical profiles may include deinterleaving packets to identify a short-term statistical profile.

[0013]   In general, in another aspect, a computer program product, disposed on a computer readable medium, includes instructions for causing a processor to receive network packets handled by a network entity and to build at least one long-term and at least one short-term statistical profile from at least one measure of the network packets that monitors data transfers, errors, or network connections. The instructions compare a short-term and a long-term statistical profile to determine whether the difference between the short-term statistical profile and the long-term statistical profile indicates suspicious network activity.

[0014]   In general, in another aspect, a method of network surveillance includes receiving packets at a virtual private network entity and statistically analyzing the received packets to determine whether the packets indicate suspicious network activity. The packets may or may not be decrypted before statistical analysis

[0015]   Advantages may include one or more of the following. Using long-term and a short-term statistical profiles from measures that monitor data transfers, errors, or network connections protects network components from

intrusion. As long-term profiles represent "normal" activity, abnormal activity may be detected without requiring an administrator to catalog each possible attack upon a network. Additionally, the ability to deinterleave packets to create multiple short-term profiles for comparison against a long-term profile enables the system to detect abnormal behavior that may be statistically ameliorated if only a single short-term profile was created.

[0016]   The scheme of communication network monitors also protects networks from more global attacks. For example, an attack made upon one network entity may cause other entities to be alerted. Further, a monitor that collects event reports from different monitors may correlate activity to identify attacks causing disturbances in more than one network entity.

[0017]   Additionally, statistical analysis of packets handled by a virtual private network enable detection of suspicious network activity despite virtual private network security techniques such as encryption of the network packets.

[0018]   Other features and advantages will become apparent from the following description, including the drawings, and from the claims.

Brief Description of the Drawings

[0019]   FIG. 1 is a diagram of network monitors deployed in an enterprise.

[0020]   FIG. 2 is a diagram of a network monitor that monitors an event stream.

[0021]   FIG. 3 is a diagram of a resource object that configures the network monitor of FIG. 2.

[0022]   FIG. 4 is a flowchart illustrating network surveillance.

[0023]   FIG. 5 is a flowchart illustrating multiple short-term statistical profiles for comparison against a single long-term statistical profile.

[0024]   FIG. 6 is a diagram of a computer platform suitable for deployment of a network monitor.

Detailed Description

[0025]   Referring to FIG. 1, an enterprise 10 includes different domains 12a-12c. Each domain 12a-12c includes one or more computers offering local

and network services that provide an interface for requests internal and external to the domain I2a-12c. Network services include features common to many network operating systems such as mail, HTTP, FTP, remote login, network file systems, finger, Kerberos, and SNMP. Some domains I2a-I2c may share trust relationships with other domains (either peer-to-peer or hierarchical). Alternatively, domains 12a-12c may operate in complete mistrust of all others, providing outgoing connections only or severely restricting incoming connections. Users may be local to a single domain or may possess accounts on multiple domains that allow them to freely establish connections throughout the enterprise 10.

[0026]   As shown, the enterprise 10 includes dynamically deployed network monitors 16a-I6f that analyze and respond to network activity and can interoperate to form an analysis hierarchy. The analysis hierarchy provides a framework for the recognition of more global threats to interdomain connectivity, including coordinated attempts to infiltrate or destroy connectivity across an entire network enterprise 10. The hierarchy includes service monitors I6a-16c, domain monitors I6d-I6e, and enterprise monitors I6f.

[0027]   Service monitors 16a-16c provide local real-time analysis of network packets (e.g., TCP/IP packets) handled by a network entity 14a-14c. Network entities include gateways, routers, firewalls, or proxy servers. A network entity may also be part of a virtual private network. A virtual private network (VPN) is constructed by using public wires to connect nodes. For example, a network could use the Internet as the medium for transporting data and use encryption and other security mechanisms to ensure that only authorized users access the network and that the data cannot be intercepted. A monitor 16a-16f can analyze packets both before and after decryption by a node of the virtual private network.

[0028]   Information gathered by a service monitor I6a-16c can be disseminated to other monitors I6a-16f, for example, via a subscription-based communication scheme. In a subscription-based scheme client monitors subscribe to receive analysis reports produced by server monitors. As a monitor 16a-16f produces analysis reports, the monitor 16a— I6f disseminates

these reports asynchronously to subscribers. Through subscription, monitors I6a-16f distributed throughout a large network are able to efficiently disseminate reports of malicious activity without requiring the overhead of synchronous polling.

[0029]   Domain monitors 16d-I6e perform surveillance over all or part of a domain 12a-12c. Domain monitors I6d-16e correlate intrusion reports disseminated by individual service monitors I6a-16c, providing a domain-wide perspective of activity (or patterns of activity). In addition to domain surveillance, domain monitors 16a-16c can reconfigure system parameters, interface with other monitors beyond a domain, and report threats against a domain 12a-12c to administrators. Domain monitors I6d-I6e can subscribe to service monitors 16a-I6c. Where mutual trust among domains 12a-I2c exists, domain monitors I6d-16e may establish peer relationships with one another. Peer-to-peer subscription allows domain monitors 16d-I6e to share analysis reports produced in other domains I2a-I2c. Domain monitors 16d-16e may use such reports to dynamically sensitize their local service monitors I6a-16c to malicious activity found to be occurring outside a domain I2a-I2c. Domain monitors I6d-I6e may also operate within an enterprise hierarchy where they disseminate analysis reports to enterprise monitors I6f for global correlation.

[0030]   Enterprise monitors 16f correlate activity reports produced across the set of monitored domains 12a-I2c. Enterprise 10 surveillance may be used where domains 12a-12c are interconnected under the control of a single organization, such as a large privately owned WAN (Wide Area Network). The enterprise 10, however, need not be stable in its configuration or centrally administered. For example, the enterprise 10 may exist as an emergent entity through new interconnections of domains I2a-I2c. Enterprise 10 surveillance is very similar to domain I2a-I2c surveillance: an enterprise monitor I6f subscribes to various domain monitors 16d-I6e, just as the domain monitors 16d-16e subscribed to various service monitors I6a-I6c. The enterprise monitor 16f (or monitors, as it would be important to avoid centralizing any analysis) focuses on network-wide threats such as Internet worm-like attacks, attacks repeated against common network services across domains, or coordinated attacks from

multiple domains against a single domain. As an enterprise monitor l6f recognizes commonalities in intrusion reports across domains (e.g., the spreading of a worm or a mail system attack repeated throughout the enterprise 10), the monitor 16f can help domains 12a-12c counter the attack and can sensitize other domains 12a-l2c to such attacks before they are affected. Through correlation and sharing of analysis reports, reports of problems found by one monitor 16a-16f may propagate to other monitors 16a-16f throughout the network. Interdomain event analysis is vital to addressing more global, information attacks against the entire enterprise 10.

[0031]   Referring to FIG. 2, each monitor 16 includes one or more analysis engines 22, 24. These engines 22, 24 can be dynamically added, deleted, and modified as necessary. In the dual-analysis configuration shown, a monitor 16 instantiation includes a signature analysis engine 22 and a statistical profiling engine 24. In general, a monitor 16 may include additional analysis engines that may implement other forms of analysis. A monitor 16 also includes a resolver 20 that implements a response policy and a resource object 32 that configures the monitor 16. The monitors 16 incorporate an application programmers' interface (API) that enhances encapsulation of monitor functions and eases integration of third-party intrusion-detection tools 28, 30.

[0032]   Each monitor 16 can analyze event records that form an event stream. The event stream may be derived from a variety of sources such as TCP/IP network packet contents or event records containing analysis reports disseminated by other monitors. For example, an event record can be formed from data included in the header and data segment of a network packet. The volume of packets transmitted and received, however, dictates careful assessment of ways to select and organize network packet information into event record streams.

[0033]   Selection of packets can be based on different criteria. Streams of event records can be derived from discarded traffic (i.e., packets not allowed through the gateway because they violate filtering rules), pass-through traffic (i.e., packets allowed into the internal network from external sources), packets having a common protocol (e.g., all ICMP (Internet. Control Message Protocol)

packets that reach the gateway), packets involving network connection management (e.g., SIN, RESET, ACK, [window resize]), and packets targeting ports to which an administrator has not assigned any network service and that also remain unblocked by the firewall. Event streams may also be based on packet source addresses (e.g., packets whose source addresses match well-known external sites such as satellite offices or have raised suspicion from other monitoring efforts) or destination addresses (e.g., packets whose destination addresses match a given internal host or workstation). Selection can also implement application-layer monitoring (e.g., packets targeting a particular network service or application). Event records can also be produced from other sources of network packet information such as report logs produced by network entities. Event streams can be of very fine granularity. For example, a different stream might be derived for commands received from different commercial web-browsers since each web-browser produces different characteristic network activity.

[0034] A monitor 16 can also construct interval summary event records, which contain accumulated network traffic statistics (e.g., number of packets and number of kilobytes transferred). These event records are constructed at the end of each interval (e.g., once per N seconds). Event records are forwarded to the analysis engines 22, 24 for analysis.

[0035] The profile engine 22 can use a wide range of multivariate statistical measures to profile network activity indicated by an event stream. A statistical score represents how closely currently observed usage corresponds to the established patterns of usage. The profiler engine 22 separates profile management and the mathematical algorithms used to assess the anomaly of events. The profile engine 22 may use a statistical analysis technique described in A. Valdes and D. Anderson, "Statistical Methods for Computer Usage Anomaly Detection Using NIDES", Proceedings of the Third International Workshop on Rough Sets and Soft Computing, January 1995, which is incorporated by reference in its entirety. Such an engine 22 can profile network activity via one or more variables called measures. Measures can be

categorized into four classes: categorical, continuous, intensity, and event distribution measures.

[0036]   Categorical measures assume values from a discrete, nonordered set of possibilities. Examples of categorical measures include network source and destination addresses, commands (e.g., commands that control data transfer and manage network connections), protocols, error codes (e.g., privilege violations, malformed service requests, and malformed packet codes), and port identifiers. The profiler engine 22 can build empirical distributions of the category values encountered, even if the list of possible values is open-ended. The engine 22 can have mechanisms for "aging out" categories whose long-term probabilities drop below a threshold.

[0037]   Continuous measures assume values from a continuous or ordinal set. Examples include inter-event time (e.g., difference in time stamps between consecutive events from the same stream), counting measures such as the number of errors of a particular type observed in the recent past, the volume of data transfers over a period of time, and network traffic measures (number of packets and number of kilobytes). The profiler engine 22 treats continuous measures by first allocating bins appropriate to the range of values of the underlying measure, and then tracking the frequency of observation of each value range. In this way, multi-modal distributions are accommodated and much of the computational machinery used for categorical measures is shared. Continuous measures are useful not only for intrusion detection, but also to support the monitoring of the health and status of the network from the perspective of connectivity and throughput. For example, a measure of traffic volume maintained can detect an abnormal loss in the data rate of received packets when this volume falls outside historical norms. This sudden drop can be specific both to the network entity being monitored and to the time of day (e.g., the average sustained traffic rate for a major network artery is much different at 11:00 a.m. than at midnight).

[0038]   Intensity measures reflect the intensity of the event stream (e.g., number of ICMP packets) over specified time intervals (e.g., 1 minute, 10

minutes, and 1 hour). Intensity measures are particularly suited for detecting flooding attacks, while also providing insight into other anomalies.

[0039]   Event distribution measures are meta-measures that describes how other measures in the profile are affected by each event. For example, an "ls" command in an FTP session affects the directory measure, but does not affect measures related to file transfer. This measure is not interesting for all event streams. For example, all network-traffic event records affect the same measures (number of packets and kilobytes) defined for that event stream, so the event distribution does not change. On the other hand, event distribution measures are useful in correlative analysis performed by a monitor l6a-16f that receives reports from other monitors l6a-l6f.

[0040]   The system maintains and updates a description of behavior with respect to these measure types in an updated profile. The profile is subdivided into short-term and long-term profiles. The short-term profile accumulates values between updates, and exponentially ages (e.g., weighs data based on how long ago the data was collected) values for comparison to the long-term profile. As a consequence of the aging mechanism, the short-term profile characterizes recent activity, where "recent" is determined by a dynamically configurable aging parameters. At update time (typically, a time of low system activity), the update function folds the short-term values observed since the last update into the long-term profile, and the short-term profile is cleared. The long-term profile is itself slowly aged to adapt to changes in subject activity. Anomaly scoring compares related attributes in the short-term profile against the long-term profile. As all evaluations are done against empirical distributions, no assumptions of parametric distributions are made, and multi-modal and categorical distributions are accommodated. Furthermore, the algorithms require no a priori knowledge of intrusive or exceptional activity.

[0041]   The statistical algorithm adjusts a short-term profile for the measure values observed in the event record. The distribution of recently observed values is compared against the long-term profile, and a distance between the two is obtained. The difference is compared to a historically adaptive deviation. The empirical distribution of this deviation is transformed to

network services. Layering signature engine analysis enables the engines 24 to avoid misguided searches along incorrect signature paths in addition to distributing the signature analysis.

[0045]   A signature engines 24 can detect, for example, address spoofing, tunneling, source routing, SATAN attacks, and abuse of ICMP messages ("Redirect" and "Destination Unreachable" messages in particular). Threshold analysis is a rudimentary, inexpensive signature analysis technique that records the occurrence of specific events and, as the name implies, detects when the number of occurrences of that event surpasses a reasonable count. For example, monitors can encode thresholds to monitor activity such as the number of fingers, pings, or failed login requests to accounts such as guest, demo, visitor, anonymous FTP, or employees who have departed the company.

[0046]   Signature engine 24 can also examine the data portion of packets in search of a variety of transactions that indicate suspicious, if not malicious, intentions by an external client. The signature engine *24,* for example, can parse FTP traffic traveling through the firewall or router for unwanted transfers of configuration or specific system data, or anonymous requests to access non-public portions of the directory structure. Similarly, a monitor can analyze anonymous FTP sessions to ensure that the file retrievals and uploads/modifications are limited to specific directories. Additionally, signature analysis capability can extend to session analyses of complex and dangerous, but highly useful, services like HTTP or Gopher.

[0047]   Signature analysis can also scan traffic directed at unused ports (i.e., ports to which the administrator has not assigned a network service). Here, packet parsing can be used to study network traffic after some threshold volume of traffic, directed at an unused port, has been exceeded. A signature engine 24 can also employ a knowledge base of known telltale packets that are indicative of well-known network-service protocol traffic (e.g., FTP, Telnet, SMTP, HTTP). The signature engine 24 then determines whether the unknown port traffic matches any known packet sets. Such comparisons could lead to the discovery of network services that have been installed without an administrator's knowledge.

[0048]   The analysis engines 22, 24 receive large volumes of events and produce smaller volumes of intrusion or suspicion reports that are then fed to the resolver 20. The resolver 20 is an expert system that receives the intrusion and suspicion reports produced by the analysis engines 22, 24 and reports produced externally by other analysis engines to which it subscribes. Based on these reports, the resolver 20 invokes responses. Because the volume of intrusion and suspicion reports is lower than the volume of events received by the analysis engines 22, 24, the resolver 20 can afford the more sophisticated demands of configuration maintenance and managing the response handling and external interfaces necessary for monitor operation. Furthermore, the resolver 20 adds to extensibility by providing the subscription interface through which third-party analysis tools 28, 30 can interact and participate in the hierarchical analysis scheme.

[0049]   Upon its initialization, the resolver 20 initiates authentication and subscription sessions with those monitors l6a-l6f whose identities appear in the monitor's 16 subscription-list (46 FIG. 3). The resolver 20 also handles all incoming requests by subscribers, which must authenticate themselves to the resolver 20. Once a subscription session is established with a subscriber monitor, the resolver 20 acts as the primary interface through which configuration requests are received and intrusion reports are disseminated.

[0050]   Thus, resolvers 20 can request and receive reports from other resolvers at lower layers in the analysis hierarchy. The resolver 20 forwards analysis reports received from subscribees to the analysis engines 22, 24. This tiered collection and correlation of analysis results allows monitors l6a—l6f to represent and profile global malicious or anomalous activity that is not visible locally.

[0051]   In addition to external-interface responsibilities, the resolver 20 operates as a fully functional decision engine, capable of invoking real-time response measures in response to malicious or anomalous activity reports produced by the analysis engines. The resolver 20 also operates as the center of intramonitor communication. As the analysis engines 22, 24 build intrusion and suspicion reports, they propagate these reports to the resolver 20 for

further correlation, response, and dissemination to other monitors l6a-l6f. The resolver 20 can also submit runtime configuration requests to the analysis engines 22, 24, for example, to increase or decrease the scope of analyses (e.g., enable or disable additional signature rules) based on various operating metrics. These configuration requests could be made as a result of encountering other intrusion reports from other subscribers. For example, a report produced by a service monitor 16a-l6c in one domain could be propagated to an enterprise monitor l6f, which in turn sensitizes service monitors in other domains to the same activity.

[0052]   The resolver 20 also operates as the interface mechanism between administrators and the monitor 16. From the perspective of a resolver 20, the administrator interface is simply a subscribing service to which the resolver 20 may submit reports and receive configuration requests. An administrative interface tool can dynamically subscribe and unsubscribe to any of the deployed resolvers 20, as well as submit configuration requests and asynchronous probes as desired.

[0053]   The monitors l6a—l6f incorporate a bidirectional messaging system that uses a standard interface specification for communication within and between monitor elements and external modules. Using this interface specification, third-party modules 28, 30 can communicate with monitors. For example, third-party modules 28 can submit event records to the analysis engines 22, 24 for processing. Additionally, third-party modules 30 may also submit and receive analysis results via the resolver's 20 external interfaces. Thus, third-party modules 28, 30 can incorporate the results from monitors into other surveillance efforts or contribute their results to other monitors l6a-16f. Lastly, the monitor's 16 internal API allows third-party analysis engines to be linked directly into the monitor boundary.

[0054]   The message system operates under an asynchronous communication model for handling results dissemination and processing that is generically referred to as subscription-based message passing. Component interoperation is client/server-based, where a client module may subscribe to receive event data or analysis results from servers. Once a subscription

request is accepted by the server, the server module forwards events or analysis results to the client automatically as data becomes available, and may dynamically reconfigure itself as requested by the client's control requests. This asynchronous model reduces the need for client probes and acknowledgments.

[0055]   The interface supports an implementation-neutral communication framework that separates the programmer's interface specification and the issues of message transport. The interface specification embodies no assumptions about implementation languages, host platform, or a network. The transport layer is architecturally isolated from the internals of the monitors so that transport modules may be readily introduced and replaced as protocols and security requirements are negotiated between module developers. The interface specification involves the definition of the messages that the various intrusion-detection modules must convey to one another and how these messages should be processed. The message structure and content are specified in a completely implementation-neutral context.

[0056]   Both intramonitor and intermonitor communication employ identical subscription-based client-server models. With respect to intermonitor communication, the resolver 20 operates as a client to the analysis engines, and the analysis engines 22, 24 operate as clients to the event filters. Through the internal message system, the resolver 20 submits configuration requests to the analysis engines 22, 24, and receives from the analysis engines 22, 24 their analysis results. The analysis engines 22, 24 operate as servers providing the resolver 20 with intrusion or suspicion reports either asynchronously or upon request. Similarly, the analysis engines 22, 24 are responsible for establishing and maintaining a communication link with an event collection method (or event filter) and prompting the reconfiguration of the collection method's filtering semantics when necessary.

[0057]   Intermonitor communication also operates using the subscription-based hierarchy. A domain monitor 16d-16e subscribes to the analysis results produced by service monitors I6a-I6c, and then propagates its own analytical reports to its parent enterprise monitor I6f. The enterprise monitor 16f operates as a client to one or more domain monitors I6d-16e, allowing them to correlate

and model enterprise-wide activity from the domain-layer results. Domain monitors I6d-I6e operate as servers to the enterprise monitors I6f, and as clients to the service monitors I6a-I6c deployed throughout their domain 12a-12c. This message scheme can operate substantially the same if correlation were to continue at higher layers of abstraction beyond enterprise 10 analysis.

[0058]   Intramonitor and intermonitor programming interfaces are substantially the same. These interfaces can be subdivided into five categories of interoperation: channel initialization and termination, channel synchronization, dynamic configuration, server probing, and report/event dissemination. Clients are responsible for initiating and terminating channel sessions with servers. Clients are also responsible for managing channel synchronization in the event of errors in message sequencing or periods of failed or slow response (i.e., "I'm alive" confirmations). Clients may also submit dynamic configuration requests to servers. For example, an analysis engine 22, 24 may request an event collection method to modify its filtering semantics. Clients may also probe servers for report summaries or additional event information. Lastly, servers may send clients intrusion/suspicion reports in response to client probes or in an asynchronous dissemination mode.

[0059]   The second part of the message system framework involves specification of a transport mechanism used to establish a given communication channel between monitors I6a-16f or possibly between a monitor I6a-I6f and a third-party security module. All implementation dependencies within the message system framework are addressed by pluggable transport modules. Transport modules are specific to the participating intrusion-detection modules, their respective hosts, and potentially to the network--should the modules require cross-platform interoperation. Instantiating a monitor I6a-I6f may involve incorporation of the necessary transport module(s) (for both internal and external communication)

[0060]   The transport modules that handle intramonitor communication may be different from the transport modules that handle intermonitor communication. This allows the intramonitor transport modules to address security and reliability issues differently than how the intermonitor transport

modules address security and reliability. While intramonitor communication may more commonly involve interprocess communication within a single host, intermonitor communication will most commonly involve cross-platform networked interoperation. For example, the intramonitor transport mechanisms may employ unnamed pipes which provides a kernel-enforced private interprocess communication channel between the monitor 16 components (this assumes a process hierarchy within the monitor 16 architecture). The monitor's 16 external transport, however, will more likely export data through untrusted network connections and thus require more extensive security management. To ensure the security and integrity of the message exchange, the external transport may employ public/private key authentication protocols and session key exchange. Using this same interface, third-party analysis tools may authenticate and exchange analysis results and configuration information in a well-defined, secure manner.

[0061]   The pluggable transport permits flexibility in negotiating security features and protocol usage with third parties. Incorporation of a commercially available network management system can deliver monitoring results relating to security, reliability, availability, performance, and other attributes. The network management system may in turn subscribe-to monitor produced results in order to influence network reconfiguration.

[0062]   All monitors (service, domain, and enterprise) 16a-16f use the same monitor code-base. However, monitors may include different resource objects 32 having different configuration data and methods. This reusable software architecture can reduce implementation and maintenance efforts. Customizing and dynamically configuring a monitor 16 thus becomes a question of building and/or modifying the resource object 32.

[0063]   Referring to FIG. 3, the resource object 32 contains the operating parameters for each of the monitor's 16 components as well as the analysis semantics (e.g., the profiler engine's 22 measure and .category definition, or the signature engine's 24 penetration rule-base) necessary to process an event stream. After defining a resource object 32 to implement a particular set of analyses on an event stream, the resource object 32 may be reused by other

monitors 16 deployed to analyze equivalent event streams. For example, the resource object 32 for a domain's router may be reused as other monitors 16 are deployed for other routers in a domain I2a-12c. A library of resource objects 32 provides prefabricated resource objects 32 for commonly available network entities.

[0064]   The resource object 32 provides a pluggable configuration module for tuning the generic monitor code-base to a specific event stream. The resource object 32 includes configurable event structures 34, analysis unit configuration 38a-38n, engine configuration 40a-40n, resolver configuration 42, decision unit configuration 44, subscription list data 46, and response methods 48.

[0065]   Configurable event structures 34 define the structure of event records and analysis result records. The monitor code-base maintains no internal dependence on the content or format of any given event stream or the analysis results produced from analyzing the event stream. Rather, the resource object 32 provides a universally applicable syntax for specifying the structure of event records and analysis results. Event records are defined based on the contents of an event stream(s). Analysis result structures are used to package the findings produced by analysis engines. Event records and analysis results are defined similarly to allow the eventual hierarchical processing of analysis results as event records by subscriber monitors.

[0066]   Event-collection methods 36 gather and parse event records for analysis engine processing. Processing by analysis engines is controlled by engine configuration 40a-40n variables and data structures that specify the operating configuration of a fielded monitor's analysis engine(s). The resource object 32 maintains a separate collection of operating parameters for each analysis engine instantiated in the monitor 16. Analysis unit configuration 38a-38n include configuration variables that define the semantics employed by the analysis engine to process the event stream.

[0067]   The resolver configuration 42 includes operating parameters that specify the configuration of the resolver's internal modules. The decision unit configuration 44 describes semantics used by the resolver's decision unit for

merging the analysis results from the various analysis engines. The semantics include the response criteria used to invoke countermeasure handlers. A resource object 32 may also include response methods 48. Response methods 48 include preprogrammed countermeasure methods that the resolver may invoke as event records are received. A response method 48 includes evaluation metrics for determining the circumstances under which the method should be invoked. These metrics include a threshold metric that corresponds to the measure values and scores produced by the profiler engine 22 and severity metrics that correspond to subsets of the associated attack sequences defined within the resource object 32.

[0068]   Countermeasures range from very passive responses, such as report dissemination to other monitors 16a—l6f or administrators, to highly aggressive actions, such as severing a communication channel or the reconfiguration of logging facilities within network components (e.g., routers, firewalls, network services, audit daemons). An active response may invoke handlers that validate the integrity of network services or other assets to ensure that privileged network services have not been subverted. Monitors 16a-16f may invoke probes in an attempt to gather as much counterintelligence about the source of suspicious traffic by using features such as traceroute or finger.

[0069]   The resource object 32 may include a subscription list 46 that includes information necessary for establishing subscription-based communication sessions, which may include network address information and public keys used by the monitor to authenticate potential clients and servers. The subscription list 46 enables transmission or reception of messages that report malicious or anomalous activity between monitors. The most obvious examples where relationships are important involve interdependencies among network services that make local policy decisions. For example, the interdependencies between access checks performed during network file system mounting and the IP mapping of the DNS service. An unexpected mount monitored by the network file system service may be responded to differently if the DNS monitor informs the network file system monitor of suspicious updates to the mount requestor's DNS mapping.

[0070]   The contents of the resource object 32 are defined and utilized during monitor 16 initialization. In addition, these fields may be modified by internal monitor 16 components, and by authorized external clients using the monitor's 16 API. Modifying the resource object 32 permits adaptive analysis of an event stream, however, it also introduces a potential stability problem if dynamic modifications are not tightly restricted to avoid cyclic modifications. To address this issue, monitors 16 can be configured to accept configuration requests from only higher-level monitors 16.

[0071]   Referring to FIG. 4, a monitor performs network surveillance by monitoring 66 a stream of network packets. The monitor builds a statistical model of network activity from the network packets, for example, by building 68 long-term and short-term statistical profiles from measures derived from the network packets. The measures include measures that can show anomalous network activity characteristic of network intrusion such as measures that describe data transfers, network connections, privilege and network errors, and abnormal levels of network traffic. The monitor can compare 70 the long-term and short-term profiles to detect suspicious network activity. Based on this comparison, the monitor can respond 72 by reporting the activity to another monitor or by executing a countermeasure response. More information can be found in P. Porras and A. Valdes "Live Traffic Analysis of TCP/IP Gateways", Networks and Distributed Systems Security Symposium, March 1998, which is incorporated by reference in its entirety.

[0072]   A few examples can illustrate this method of network surveillance. Network intrusion frequently causes large data transfers, for example, when an intruder seeks to download sensitive files or replace system files with harmful substitutes. A statistical profile to detect anomalous data transfers might include a continuous measure of file transfer size, a categorical measure of the source or destination directory of the data transfer, and an intensity measure of commands corresponding to data transfers (e.g., commands that download data). These measures can detect a wide variety of data transfer techniques such as a large volume of small data transfers via e-mail or downloading large files en masse. The monitor may distinguish between network packets based

on the time such packets were received by the network entity, permitting statistical analysis to distinguish between a normal data transfer during a workday and an abnormal data transfer on a weekend evening.

[0073]   Attempted network intrusion may also produce anomalous levels of errors. For example, categorical and intensity measures derived from privilege errors may indicate attempts to access protected files, directories, or other network assets. Of course, privilege errors occur during normal network operation as users mistype commands or attempt to perform an operation unknowingly prohibited. By comparing the long-term and short-term statistical profiles, a monitor can distinguish between normal error levels and levels indicative of intrusion without burdening a network administrator with the task of arbitrarily setting an unvarying threshold. Other measures based on errors, such as codes describing why a network entity rejected a network packet enable a monitor to detect attempts to infiltrate a network with suspicious packets.

[0074]   Attempted network intrusion can also be detected by measures derived from network connection information. For example, a measure may be formed from the correlation (e.g., a ratio or a difference) of the number of SYN connection request messages with the number of SIN_ACK connection acknowledgment messages and/or the number of ICMP messages sent. Generally, SIN requests received should balance with respect to the total of SIN_ACK and ICMP messages sent. That is, flow into and out-of a network entity should be conserved. An imbalance can indicate repeated unsuccessful attempts to connect with a system, perhaps corresponding to a methodical search for an entry point to a system. Alternatively, intensity measures of transport-layer connection requests, such as a volume analysis of SYN-RST messages, could indicate the occurrence of a SIN-attack against port availability or possibly port-scanning. Variants of this can include intensity measures of TCP/FIN messages, considered a more stealthy form of port scanning.

[0075]   Many other measures can detect network intrusion. For example, "doorknob rattling," testing a variety of potentially valid commands to gain access (e.g., trying to access a "system" account with a password of "system"),

can be detected by a variety of categorical measures. A categorical measure of commands included in network packets can identify an unusual short-term set of commands indicative of "doorknob-rattling." Similarly, a categorical measure of protocol requests may also detect an unlikely mix of such requests.

[0076]   Measures of network packet volume can also help detect malicious traffic, such as traffic intended to cause service denials or perform intelligence gathering, where such traffic may not necessarily be violating filtering policies. A measure reflecting a sharp increase in the overall volume of discarded packets as well as a measure analyzing the disposition of the discarded packets can provide insight into unintentionally malformed packets resulting from poor line quality or internal errors in neighboring hosts. High volumes of discarded packets can also indicate more maliciously intended transmissions such as scanning of UPD ports or IP address scanning via ICMP echoes. Excessive number of mail expansion request commands (EXPN) may indicate intelligence gathering, for example, by spammers.

[0077]   A long-term and short-term statistical profile can be generated for each event stream. Thus, different event streams can "slice" network packet data in different ways. For example, an event stream may select only network packets having a source address corresponding to a satellite office. Thus, a long-term and short-term profile will be generated for the particular satellite office. Thus, although a satellite office may have more privileges and should be expected to use more system resources than other external addresses, a profile of satellite office use can detect "address spoofing" (i.e., modifying packet information to have a source address of the satellite office).

[0078]   The same network packet event may produce records in more than one event stream. For example, one event stream may monitor packets for FTP commands while another event stream monitors packets from a particular address. In this case, an FTP command from the address would produce an event record in each stream.

[0079]   Referring to FIG. 5, a monitor may also "deinterleave." That is, the monitor may create and update 74, 76 more than one short-term profile for comparison 78 against a single long-term profile by identifying one of the

multiple short-term profiles that will be updated by an event record in an event stream. For example, at any one time a network entity may handle several FTP "anonymous" sessions. If each network packet for all anonymous sessions were placed in a single short-term statistical profile, potentially intrusive activity of one anonymous session may be statistically ameliorated by non-intrusive sessions. By creating and updating short-term statistical profiles for each anonymous session, each anonymous session can be compared against the long-term profile of a normal FTP anonymous session. Deinterleaving can be done for a variety of sessions including HTTP sessions (e.g., a short-term profile for each browser session).

[0080]   Referring to FIG. 6, a computer platform 14 suitable for executing a network monitor 16 includes a display 50, a keyboard 54, a pointing device 58 such as a mouse, and a digital computer 56. The digital computer 56 includes memory 62, a processor 60, a mass storage device 64a, and other customary components such as a memory bus and peripheral bus. The platform 14 may further include a network connection 52.

[0081]   Mass storage device 64a can store instructions that form a monitor 16. The instructions may be transferred to memory 62 and processor 60 in the course of operation. The instructions 16 can cause the display 50 to display images via an interface such as a graphical user interface. Of course, instructions may be stored on a variety of mass storage devices such as a floppy disk 64b, CD-ROM 640, or PROM (not shown).

[0082]   Other embodiments are within the scope of the following claims.